

XML et XQuery

XML XQUERY

XQuery : Influences

- ◆ *Quilt* : langage de requêtes pour XML
- ◆ *SQL* : Modèle « Select from where »
- ◆ *Xpath 1.0* (*) et *XQL* : Expressions de chemins
- ◆ *XML-QL* : « binding » de variables
- ◆ *ODMG/OQL* : aspect fonctionnel du langage avec expressions imbriquées

Xquery

- « Drafts » : juin 2001, avril 2002, *mai 2003*
- Principaux types d'expression XQuery
 - ◆ Expressions de chemin
 - ◆ Expressions For Let Where Return (*FLoWeR*)
 - ◆ Expressions avec opérateurs et fonctions
 - ◆ Expressions conditionnelles
 - ◆ Expressions avec quantificateurs
 - ◆ Expressions testant ou modifiant les types de données

- *Expressions de chemins*: prédicats filtrant un ensemble de nœuds
 - ◆ document(“fichier.xml”) : nœud racine
 - ◆ . équivalent à self::node()
 - ◆ .. équivalent à parent::node()
 - ◆ nom équivalent à child::nom (/)
 - ◆ @nom équivalent à attribut::nom
 - ◆ // équivalent à /descendant ou self::node()/

- ***document("zoo.xml")//chapter[2]//figure[capti
on = "Tree Frogs"]***
 - ◆ Noeud racine : zoo.xml
 - ◆ 2ème descendant du noeud racine : <chapter[2]>
 - ◆ Recherche des éléments <figure> n'importe où dans l'élément <chapter>
 - ◆ Sélectionner ceux ayant un attribut <caption> avec la valeur "Tree Frogs".
- ***document("zoo.xml")//chapter[2 TO 5]//figure***

● Chemins de localisation

Syntaxe : **indicateur de relation::filtre[prédicat]**

- ◆ Troisième enfant : **child::*[3]**
- ◆ Dernier descendant:
descendant::[position(=last0)]
- ◆ Films sans acteur : **film[count(child::acteur)=0]**
- ◆ Film dont l'attribut titre est égal à Brazil :
film[child::@titre='Brazil']

```
<DesLivres>
<book year = "1994">
  <title>TCP/IP Illustrated</title>
  <author> <last id = 12>Stevens</last>
    <first>W.</first> </author>
  <publisher>Addison-Wesley</publisher>
</book>

<book year = "1998">
  <title>TCP/IP Illustrated< /title>
  <author> <last id = 12>Stevens</last>
    <first>W.</first><</author>
  <publisher> Morgan Kaufmann </publisher>
< /book>
</DesLivres>
```

- **[FOR | LET] ... WHERE ... RETURN**
- ◆ ***FOR \$b IN document("bib.xml")//book***
WHERE \$b/publisher = "Morgan Kaufmann"
AND \$b/year = "1998"
RETURN \$b/title
- ◆ ***LET \$b := document("bib.xml")//book***
RETURN \$b/title
- **[unordered()|distinct()]**

- Quantificateurs : exemple

```
<items>
  <item_tuple>
    <itemno>1001</itemno>
    <description>Red Bicycle</description>
  </item_tuple>
  <item_tuple>
    <itemno>1002</itemno>
    <description>Red Bicycle</description>
  </item_tuple>
  <item_tuple>
    <itemno>1003</itemno>
    <description>Red Bicycle</description>
  </item_tuple>
</items>
```

● Quantificateurs: SOME | EVERY ... IN ... SATISFIES

FOR \$i *IN* document("data/R-items.xml")//item_tuple
WHERE NOT(
 SOME \$b *IN* document("data/R-bids.xml")//bid_tuple
 SATISFIES \$b/itemno = \$i/itemno)

RETURN

</no_bid_item>

{\$i/description}

</no_bid_item>

- Jointure

```
FOR  $\$i$  IN document("catalog.xml")//item,  
     $\$p$  IN document("parts.xml")//part[partno =  $\$i$ /partno],  
     $\$s$  IN document("suppliers.xml")//supplier[suppno =  
         $\$i$ /suppno]
```

```
RETURN
```

```
-----
```